

Figure 1

AT9-98-920

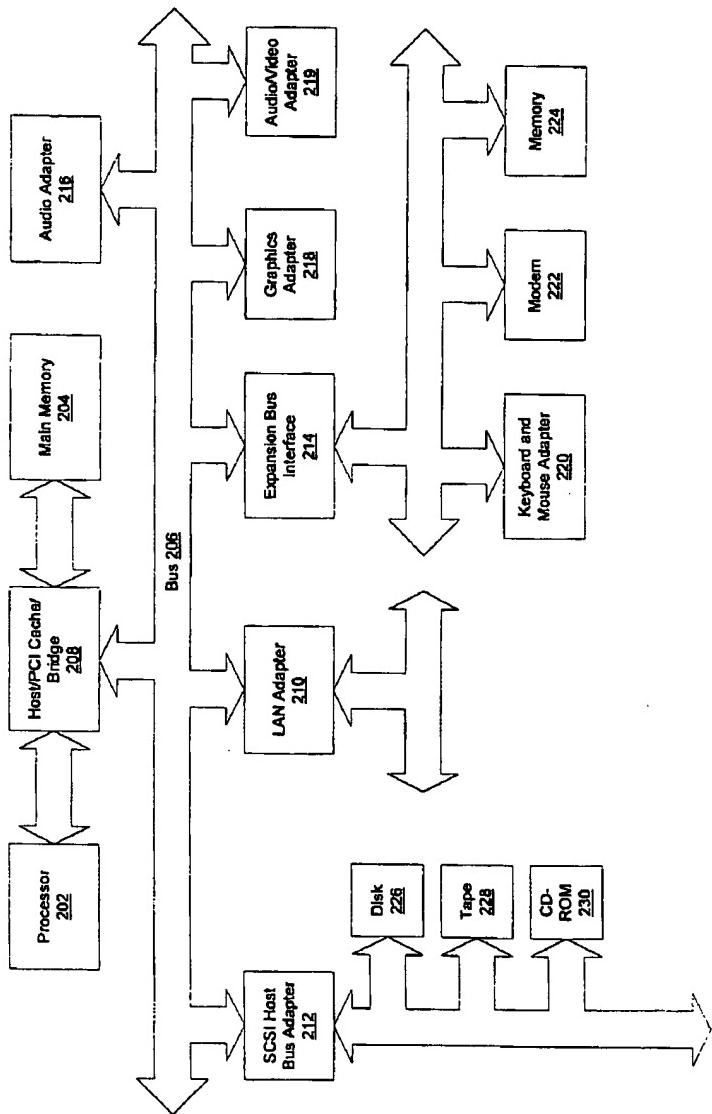
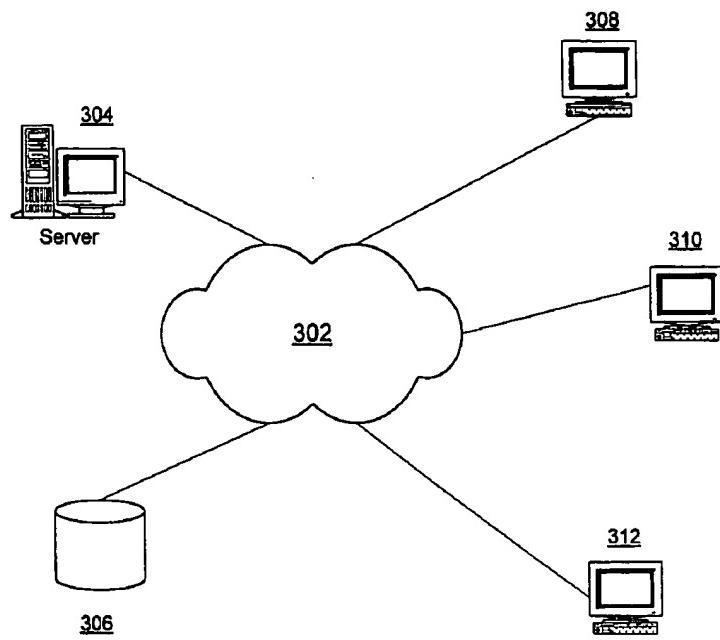


Figure 2
AT9-98-920



300
Network
Figure 3

AT9-98-920

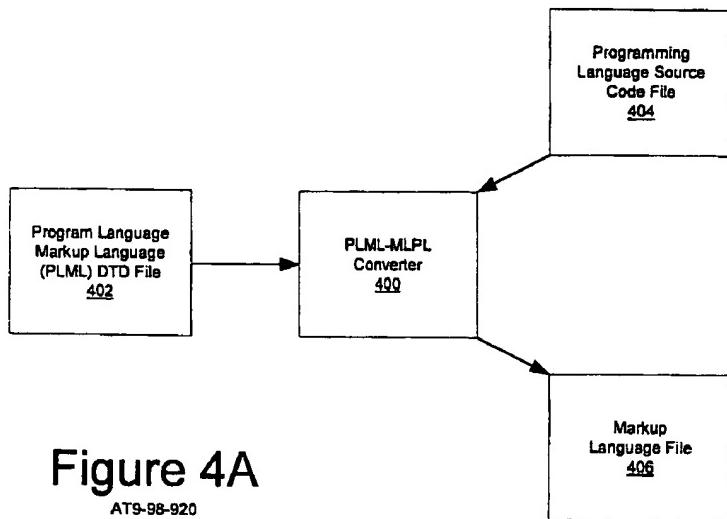


Figure 4A

AT9-98-920

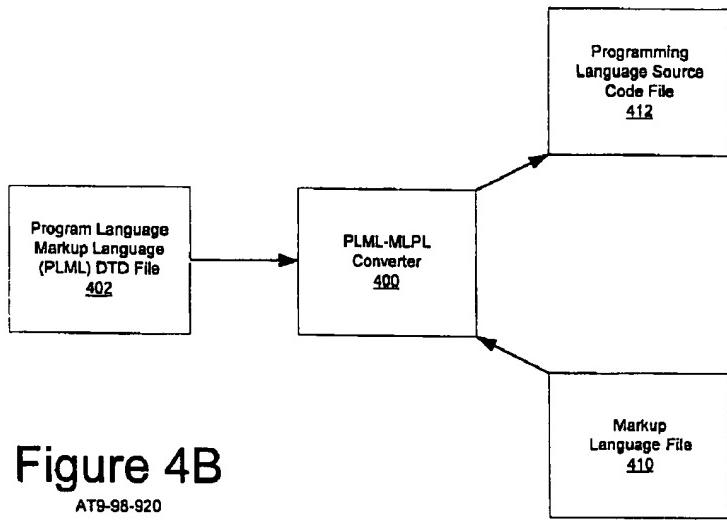


Figure 4B

AT9-98-920

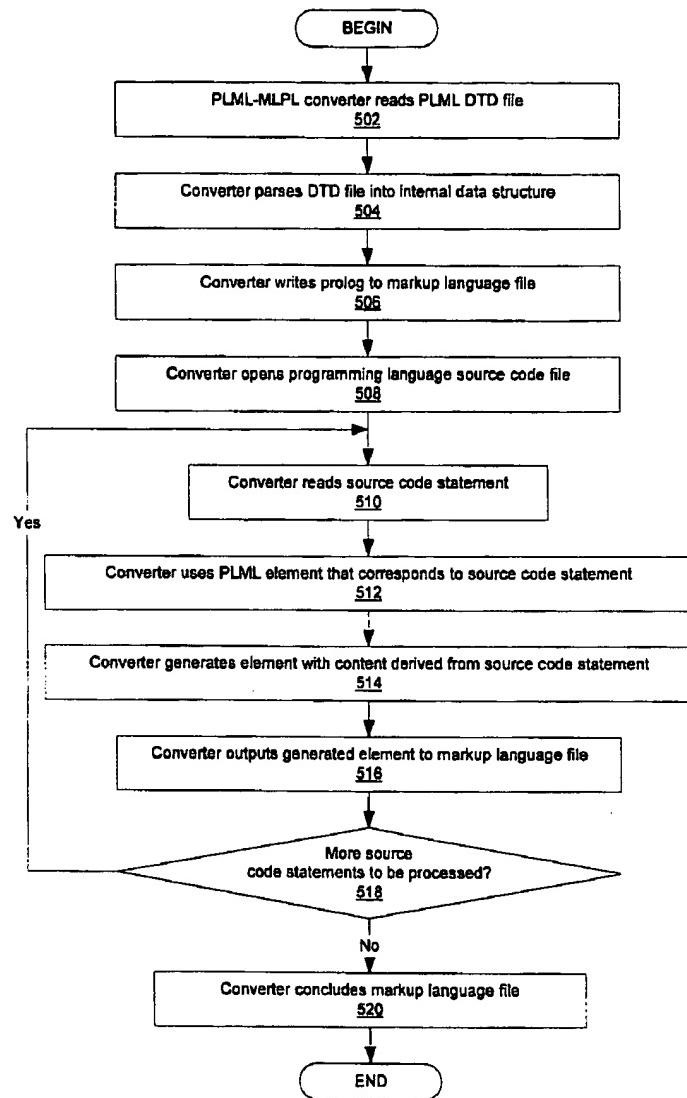


Figure 5

AT9-98-920

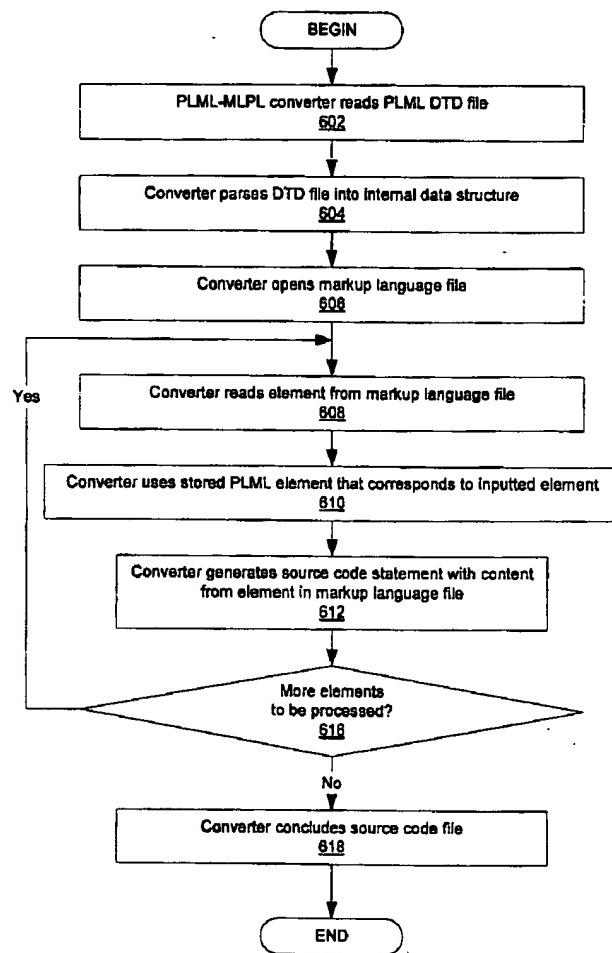


Figure 6

AT9-98-920

702 { <!ENTITY % base_content_model '(functionA | functionB)''>

704 { <!ELEMENT pml % base_content_model;>

706 { <!ELEMENT functionA EMPTY>
<!ATTLIST functionA arg1 CDATA #REQUIRED
arg2 CDATA #REQUIRED
>

708 { <!ELEMENT functionB EMPTY>
<!ATTLIST functionB arg1 CDATA #REQUIRED
>
<!-- End of DTD for Programming Language Markup Language-->

Figure 7

AT9-98-920

800 {

802 { main programA (){
integer temp;
initProg ();

804 { temp = functionA (5,7);

806 { temp = functionB (25);
}

Figure 8

AT9-98-920

902 { < ? pml version = "1.0"?>
<!DOCTYPE pml SYSTEM "pml.dtd">

904 { <pml>

906 { <!-- main programA () -->
<!-- integer temp; -->
<!-- initProg (); -->

908 { <functionA arg1="5" arg2="7" />

910 { <functionB arg1="25" />

912 { <!-- } -->

914 { </pml>

Figure 9A

AT9-98-920

922 { < ? pml version = "1.0"?>
<!DOCTYPE pml SYSTEM "pml.dtd">

924 { <pml>

926 { <functionA arg1="5" arg2="7" />

928 { <functionB arg1="25" />

930 { </pml>

Figure 9B

AT9-98-920

902 { < ? plml version = "1.0"?>
<!DOCTYPE plml SYSTEM "plml.dtd">

904 { <plml>

906 { <!-- main programA () -->
<!-- integer temp; -->
<!-- initProg (); -->

908 { < functionA arg1="5" arg2="7" />

910 { < functionB arg1="25" />

912 { <!-- -->

914 { </ plml >

Figure 9A

AT9-98-920

922 { < ? plml version = "1.0"?>
<!DOCTYPE plml SYSTEM "plml.dtd">

924 { <plml>

926 { < functionA arg1="5" arg2="7" />

928 { < functionB arg1="25" />

930 { </plml >

Figure 9B

AT9-98-920

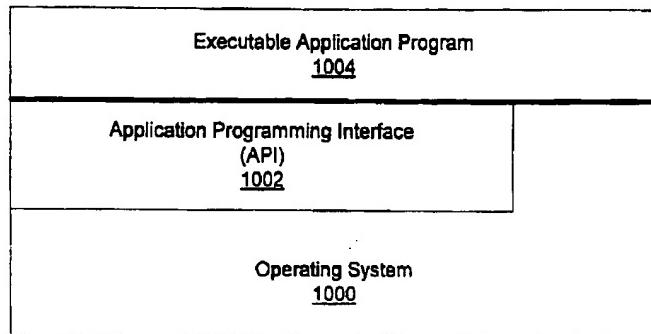


Figure 10A

AT9-98-920

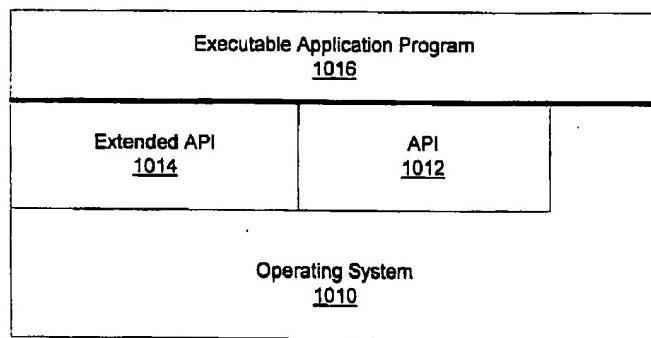


Figure 10B

AT9-98-920

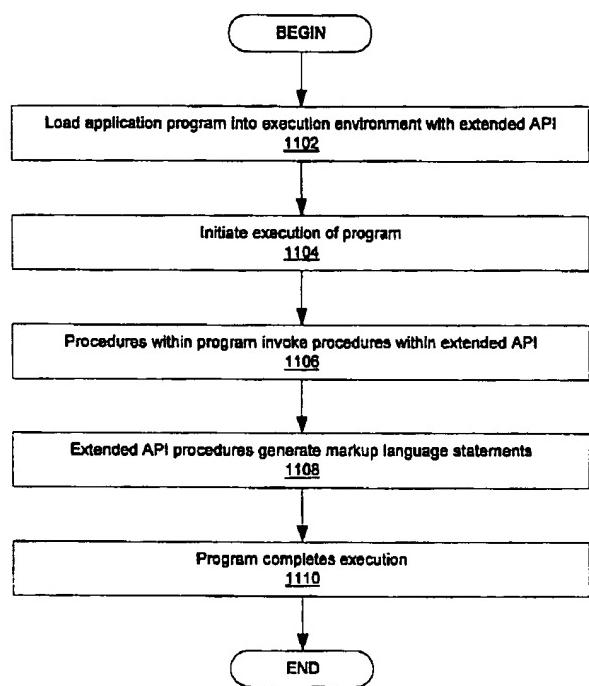


Figure 11

AT9-98-920

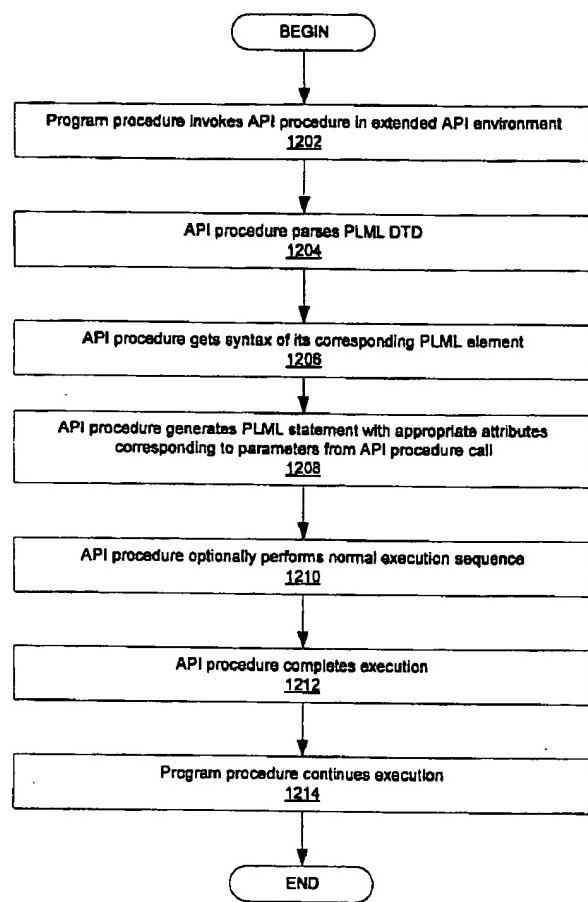


Figure 12

AT9-98-920

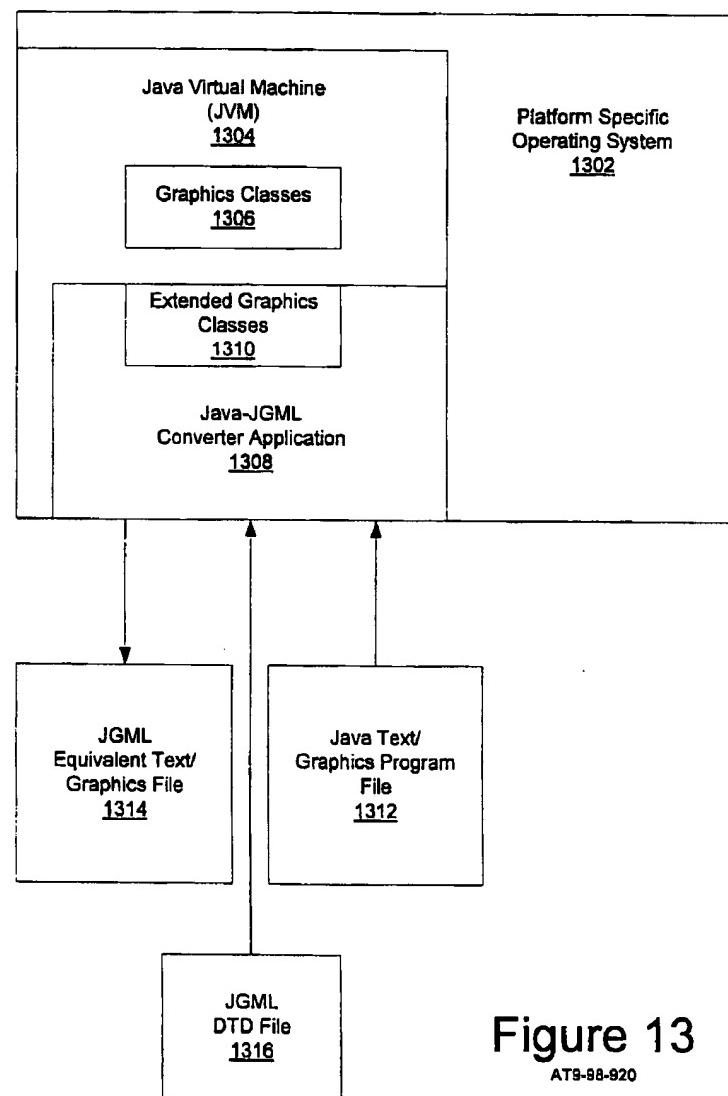


Figure 13

AT9-98-920

1400 {

1402 { public class JGML Graphics extends Graphics

 public void drawLine (int x1, int y1, int x2, int y2)

 {

1404 {

1406 { Analyze JGML DTD for "drawLine" syntax

1408 { Generate JGML output statement with "drawLine" syntax and current parameters

1410 { printLine ("<drawLine x1=" + x1 + " y1=" + y1 + " x2=" + x2
 + " y2=" + y2 + ">");

 }

1412 {

 public void clearRect(int x, int y, int width, int height)

 {

 Analyze JGML DTD for "clearRect" syntax

 Generate JGML output statement with "clearRect" syntax and current parameters

 printLine (

 <clearRect x=" + x + " y=" + y + " width=" + width + " height=" + height + ">);

 }

Figure 14

AT9-98-920

```
<!-- Java Graphics Markup Language (JGML) Document Type Definition (DTD) -->
<!ENTITY % base_content_model
  '(copyArea | drawLine | fillRect | drawRect | clearRect |
    drawRoundRect | fillRoundRect | draw3Drect | fill3Drect |
    drawOval | fillOval | drawArc | fillArc | drawPolyline |
    drawPolygon | fillPolygon | drawString | drawChars |
    drawBytes | drawImage | dispose | finalize | clipRect |
    setClip | setColor | setPaintMode | translate | setXORMode |
    setFont)*'
>
<!ELEMENT jgml %base_content_model;>
<!ELEMENT copyArea          EMPTY>
<!ATTLIST
  copyArea      x      CDATA      #REQUIRED
                y      CDATA      #REQUIRED
                width   CDATA      #REQUIRED
                height  CDATA      #REQUIRED
                dx     CDATA      #REQUIRED
                dy     CDATA      #REQUIRED
>
<!ELEMENT drawLine          EMPTY>
<!ATTLIST
  drawLine      x1    CDATA      #REQUIRED
                y1    CDATA      #REQUIRED
                x2    CDATA      #REQUIRED
                y2    CDATA      #REQUIRED
>
<!ELEMENT fillRect          EMPTY>
<!ATTLIST
  fillRect      x      CDATA      #REQUIRED
                y      CDATA      #REQUIRED
                width  CDATA      #REQUIRED
                height CDATA      #REQUIRED
>
<!ELEMENT drawRect          EMPTY>
<!ATTLIST
  drawRect      x      CDATA      #REQUIRED
                y      CDATA      #REQUIRED
                width  CDATA      #REQUIRED
                height CDATA      #REQUIRED
>
<!ELEMENT clearRect         EMPTY>
<!ATTLIST
  clearRect     x      CDATA      #REQUIRED
                y      CDATA      #REQUIRED
                width  CDATA      #REQUIRED
                height CDATA      #REQUIRED
>
```

Figure 15A

AT9-98-920

```
<!ELEMENT drawRoundRect    EMPTY>
<!ATTLIST
  drawRoundRect
    x          CDATA      #REQUIRED
    y          CDATA      #REQUIRED
    width     CDATA      #REQUIRED
    height    CDATA      #REQUIRED
    arcWidth  CDATA      #REQUIRED
    arcHeight CDATA      #REQUIRED
  >
<!ELEMENT fillRoundRect   EMPTY>
<!ATTLIST
  fillRoundRect
    x          CDATA      #REQUIRED
    y          CDATA      #REQUIRED
    width     CDATA      #REQUIRED
    height    CDATA      #REQUIRED
    arcWidth  CDATA      #REQUIRED
    arcHeight CDATA      #REQUIRED
  >
<!ELEMENT draw3DRect     EMPTY>
<!ATTLIST
  draw3DRect
    x          CDATA      #REQUIRED
    y          CDATA      #REQUIRED
    width     CDATA      #REQUIRED
    height    CDATA      #REQUIRED
    raised    CDATA      #REQUIRED
  >
<!ELEMENT fill3DRect     EMPTY>
<!ATTLIST
  fill3DRect
    x          CDATA      #REQUIRED
    y          CDATA      #REQUIRED
    width     CDATA      #REQUIRED
    height    CDATA      #REQUIRED
    raised    CDATA      #REQUIRED
  >
<!ELEMENT drawOval        EMPTY>
<!ATTLIST
  drawOval
    x          CDATA      #REQUIRED
    y          CDATA      #REQUIRED
    width     CDATA      #REQUIRED
    height    CDATA      #REQUIRED
  >
<!ELEMENT fillOval        EMPTY>
<!ATTLIST
  fillOval
    x          CDATA      #REQUIRED
    y          CDATA      #REQUIRED
    width     CDATA      #REQUIRED
    height    CDATA      #REQUIRED
  >
```

Figure 15B

AT9-98-920

```
<!ELEMENT drawArc          EMPTY>
<!ATTLIST
  drawArc
    x      CDATA #REQUIRED
    y      CDATA #REQUIRED
    width CDATA #REQUIRED
    height CDATA #REQUIRED
    startAngle CDATA #REQUIRED
    arcAngle  CDATA #REQUIRED
  >
<!ELEMENT fillArc          EMPTY>
<!ATTLIST
  fillArc
    x      CDATA #REQUIRED
    y      CDATA #REQUIRED
    width CDATA #REQUIRED
    height CDATA #REQUIRED
    startAngle CDATA #REQUIRED
    arcAngle  CDATA #REQUIRED
  >
<!ELEMENT drawPolyLine     EMPTY>
<!ATTLIST
  drawPolyLine
    xPoints CDATA #REQUIRED
    yPoints CDATA #REQUIRED
    nPoints CDATA #REQUIRED
  >
<!ELEMENT drawPolygon      EMPTY>
<!ATTLIST
  drawPolygon
    xPoints CDATA #IMPLIED
    yPoints CDATA #IMPLIED
    nPoints CDATA #IMPLIED
    p       CDATA #IMPLIED
  >
<!ELEMENT fillPolygon      EMPTY>
<!ATTLIST
  fillPolygon
    xPoints CDATA #IMPLIED
    yPoints CDATA #IMPLIED
    nPoints CDATA #IMPLIED
    Polygon CDATA #IMPLIED
  >
<!ELEMENT drawString       EMPTY>
<!ATTLIST
  drawString
    str    CDATA #REQUIRED
    x     CDATA #REQUIRED
    y     CDATA #REQUIRED
  >
```

Figure 15C

AT9-98-920

```
<!ELEMENT drawChars          EMPTY>
<!ATTLIST
  drawChars
    data      CDATA #REQUIRED
    offset    CDATA #REQUIRED
    length   CDATA #REQUIRED
    x        CDATA #REQUIRED
    y        CDATA #REQUIRED
  >
<!ELEMENT drawBytes          EMPTY>
<!ATTLIST
  drawBytes
    offset    CDATA #REQUIRED
    length   CDATA #REQUIRED
    x        CDATA #REQUIRED
    y        CDATA #REQUIRED
  >
<!ELEMENT drawImage          EMPTY>
<!ATTLIST
  drawImage
    img      CDATA #REQUIRED
    x       CDATA #IMPLIED
    y       CDATA #IMPLIED
    width   CDATA #IMPLIED
    height  CDATA #IMPLIED
    dx1     CDATA #IMPLIED
    dy1     CDATA #IMPLIED
    dx2     CDATA #IMPLIED
    dy2     CDATA #IMPLIED
    sx1     CDATA #IMPLIED
    sy1     CDATA #IMPLIED
    sx2     CDATA #IMPLIED
    sy2     CDATA #IMPLIED
    bgcolor CDATA #IMPLIED
    observer CDATA #REQUIRED
  >
<!ELEMENT dispose            EMPTY>
<!ELEMENT finalize           EMPTY>
<!ELEMENT clipRect           EMPTY>
<!ATTLIST
  clipRect
    x        CDATA #REQUIRED
    y        CDATA #REQUIRED
    width   CDATA #REQUIRED
    height  CDATA #REQUIRED
  >
```

Figure 15D

AT9-98-920

```
<!ELEMENT setClip          EMPTY>
<!ATTLIST
  setClip      x      CDATA    #IMPLIED
               y      CDATA    #IMPLIED
               width   CDATA    #IMPLIED
               height  CDATA    #IMPLIED
               clip    CDATA    #IMPLIED
>
<!ELEMENT setColor         EMPTY>
<!ATTLIST
  setColor     color   CDATA    #REQUIRED
<!ELEMENT setPaintmode    EMPTY>
<!ELEMENT translate       EMPTY>
<!ATTLIST
  translate    x      CDATA    #REQUIRED
               y      CDATA    #REQUIRED
>
<!ELEMENT setXORMode      EMPTY>
<!ATTLIST
  setXORMode   c1     CDATA    #REQUIRED
>
<!ELEMENT setFont          EMPTY>
<!ATTLIST
  setFont      font   CDATA    #REQUIRED
>
<!-- End of DTD for Java Graphics Markup Language -->
```

Figure 15E

AT9-98-920

- **clearRect** (int, int, int, int)
Clears the specified rectangle by filling it with the background color of the current drawing surface.
- **clipRect** (int, int, int, int)
Intersects the current clip with the specified rectangle.
- **copyArea** (int, int, int, int, int, int)
Copies an area of the component by a distance specified by dx and dy.
- **create** ()
Creates a new Graphics object that is a copy of the Graphics object.
- **create** (int, int, int, int)
Creates a new Graphics object based on this Graphics object, but with a new translation and clip area.
- **dispose** ()
Disposes of this graphics context and releases any system resources that it is using.
- **draw3DRect** (int, int, int, int, boolean)
Draws a 3-D highlighted outline of the specified rectangle.
- **drawArc** (int, int, int, int, int, int)
Draws the outline of a circular or elliptical arc covering the specified rectangle.
- **drawBytes** (byte[], int, int, int, int)
Draws the text given by the specified byte array, using this graphics context's current font and color.
- **drawChars** (char[], int, int, int, int)
Draws the text given by the specified character array, using this graphics context's current font and color.
- **drawImage** (Image, int, int, Color, ImageObserver)
Draws as much of the specified image as is currently available.
- **drawImage** (Image, int, int, int, int, Color, ImageObserver)
Draws as much of the specified image as has already been scaled to fit inside the specified rectangle.
- **drawImage** (Image, int, int, int, int, int, ImageObserver)
Draws as much of the specified image as has already been scaled to fit inside the specified rectangle.
- **drawImage** (Image, int, int, int, int, int, int, int, Color, ImageObserver)
Draws as much of the specified area of the specified image as is currently available, scaling it on the fly to fit inside the specified area of the destination drawable surface.
- **drawImage** (Image, int, int, int, int, int, int, int, int, int, ImageObserver)
Draws as much of the specified area of the specified image as is currently available, scaling it on the fly to fit inside the specified area of the destination drawable surface.
- **drawLine** (int, int, int, int)
Draws a line, using the current color, between the points (x1, y1) and (x2, y2) in this graphics context's coordinate system.
- **drawOval** (int, int, int, int)
Draws the outline of an oval.
- **drawPolygon** (int[], int[], int)
Draws a closed polygon defined by arrays of x and y coordinates.
- **drawPolygon** (Polygon)
Draws the outline of a polygon defined by the specified Polygon object.
- **drawPolyline** (int[], int[], int)
Draws a sequence of connected lines defined by arrays of x and y coordinates.
- **drawRect** (int, int, int, int)
Draws the outline of the specified rectangle.
- **drawRoundRect** (int, int, int, int, int, int)
Draws an outlined round-cornered rectangle using this graphics context's current color.

Figure 16A

AT9-98-920

- **drawString** (*String, int, int*)
Draws the text given by the specified string, using this graphics context's current font and color.
- **fill3DRect** (*int, int, int, int, boolean*)
Paints a 3-D highlighted rectangle filled with the current color.
- **fillArc** (*int, int, int, int, int, int*)
Fills a circular or elliptical arc covering the specified rectangle.
- **fillOval** (*int, int, int, int*)
Fills an oval bounded by the specified rectangle with the current color.
- **fillPolygon** (*int[], int[], int*)
Fills a closed polygon defined by arrays of x and y coordinates.
- **fillPolygon** (*Polygon*)
Fills the polygon defined by the specified Polygon object with the graphics context's current color.
- **fillRect** (*int, int, int, int*)
Fills the specified rectangle.
- **fillRoundRect** (*int, int, int, int, int, int*)
Fills the specified rounded corner rectangle with the current color.
- **finalize** ()
Disposes of this graphics context once it is no longer referenced.
- **getClip** ()
Gets the current clipping area.
- **getClipBounds** ()
Returns the bounding rectangle of the current clipping area.
- **getClipRect** ()
Deprecated.
- **getColor** ()
Gets this graphics context's current color.
- **getFont** ()
Gets the current font.
- **getFontMetrics** ()
Gets the font metrics of the current font.
- **getFontMetrics** (*Font*)
Gets the font metrics for the specified font.
- **setClip** (*int, int, int, int*)
Sets the current clip to the rectangle specified by the given coordinates.
- **setClip** (*Shape*)
Sets the current clipping area to an arbitrary clip shape.
- **setColor** (*Color*)
Sets this graphics context's current
- **setFont** (*Font*)
Sets this graphics context's font to the specified font.
- **setPaintMode** ()
Sets the paint mode of this graphics context to overwrite the destination with this graphics context's current color.
- **setXORMode** (*Color*)
Sets the paint mode of this graphics context to alternate between this graphics context's current color and the new specified color.
- **toString** ()
Returns a String object representing this Graphics object's value.
- **translate** (*int, int*)
Translates the origin of the graphics context to the point (x, y) in the current coordinate system.

Figure 16B

AT9-98-920

1702 { < ! ELEMENT drawLine EMPTY>
1706 { < ! ATTLIST drawLine x1 CDATA #REQUIRED
x2 CDATA #REQUIRED
y1 CDATA #REQUIRED
y2 CDATA #REQUIRED
>
1700 {
1704 { < ! ELEMENT clearRect EMPTY>
1708 { < ! ATTLIST clearRect x CDATA #REQUIRED
y CDATA #REQUIRED
width CDATA #REQUIRED
height CDATA #REQUIRED
>

Figure 17

AT9-98-920

1800 {
1802 - drawLine (23, 43, 50, 60);
1804 - drawLine (50, 60, 27, 80);
1806 - clearRect (0, 0, 10, 10);

Figure 18

AT9-98-920

1900 {

```
< ? xml version="1.0" ? >
<!DOCTYPE jgml SYSTEM "jgml.dtd" >
<jgml>.
1902 - < drawLine x1="23" y1="43" x2="50" y2="60" / >
1904 - < drawLine x1="50" y1="60" x2="27" y2="80" / >
1906 - < clearRect x="0" y="0" width="10" height="10" / >
          </jgml>
```

Figure 19

AT9-98-920